

Simultaneous Perturbation Algorithms for Batch Off-Policy Search

Raphael Fonteneau ^{*1} and Prashanth L A ^{†2}

¹Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium

²INRIA Lille - Nord Europe, Team SequeL, FRANCE.

Abstract

We propose novel policy search algorithms in the context of off-policy, batch mode reinforcement learning (RL) with continuous state and action spaces. Given a batch collection of trajectories, we perform off-line policy evaluation using an algorithm similar to that by Fonteneau et al. [2010]. Using this Monte-Carlo like policy evaluator, we perform policy search in a class of parameterized policies. We propose both first order policy gradient and second order policy Newton algorithms. All our algorithms incorporate simultaneous perturbation estimates for the gradient as well as the Hessian of the cost-to-go vector, since the latter is unknown and only biased estimates are available. We demonstrate their practicality on a simple 1-dimensional continuous state space problem.

1 Introduction

This paper stands within the field of optimal control in the context of infinite horizon discounted cost Markov decision processes (MDPs) Bertsekas and Tsitsiklis [1996]. More specifically, this paper addresses the batch mode setting Ernst et al. [2005], Fonteneau [2011], where we are given a set of noisy trajectories of a system without access to any model or simulator of that system. More formally, we are given a set of n samples (also called transitions) $\{(x^l, u^l, c^l, y^l)\}_{l=1}^n$, where, for every $l \in \{1, \dots, n\}$, the 4-tuple (x^l, u^l, c^l, y^l) denotes the state x^l , the action u^l , a (noisy) cost received in (x^l, u^l) and a (noisy) successor state reached when taking action u^l in state x^l . The samples are generated according to some unknown policy and the objective is to develop a (off-policy) control scheme that attempts to find a near-optimal policy using this batch of samples.

For this purpose, we first parameterize the policy and hence the cost-to-go, denoted by $J^\theta(x_0)$. Here θ is the policy parameter, x_0 is a given initial state and $J^\theta(x_0)$ is the expected cumulative discounted sum of costs under a policy governed by θ (see (1)). Note that the policy parameterization is not constrained to be linear. We develop algorithms that perform descent using estimates of the cost-to-go $J^\theta(x_0)$. For obtaining these estimates from the batch data, we extend a recent algorithm proposed for finite horizon MDPs Fonteneau et al. [2010], to the infinite horizon, discounted setting. The advantage of this estimator, henceforth referred to as MFMC, is that it is off-policy in nature, computationally tractable and consistent under Lipschitz assumption on the transition dynamics, cost function and policy. Moreover, it does not require the use of function approximators, but only needs a metric on the state and action spaces.

Being equipped with the MFMC policy evaluator that outputs an estimate of the cost-to-go $J^\theta(x_0)$ for any policy parameter θ , the requirement is for a control scheme that uses these estimated values to update the parameter θ in the negative descent direction. However, closed form expressions of the gradient/Hessian of the cost-to-go are not available and MFMC estimates possess a non-zero bias. To alleviate this, we employ the well-known

*raphael.fonteneau@ulg.ac.be

†prashanth.la@inria.fr

simultaneous perturbation principle (cf. Bhatnagar et al. [2013]) to estimate the gradient and Hessian, respectively, of $J^\theta(x_0)$ using estimates from MFMC and propose two first order and two second order algorithms. Our algorithms are based on two popular simultaneous perturbation methods - Simultaneous Perturbation Stochastic Approximation (SPSA) Spall [1992] and Smoothed Functional Katkovnik and Kulchitsky [1972].

The first-order algorithms perform gradient descent using either SPSA or SF estimates to update the policy parameter. On the other hand, the second order algorithms incorporate a Newton step by estimating the gradient as well as the Hessian of the cost-to-go $J^\theta(x_0)$ using SPSA or SF. We demonstrate the empirical usefulness of our algorithms on a simple 1-dimensional continuous state space problem.

To the best of our knowledge, the algorithms presented in this paper are the first to solve batch, off-policy stochastic control in continuous state and action spaces without using function approximators for evaluating policies. Our approach only requires (i) a (random) set of trajectories, (ii) metrics on the state and action spaces, and (iii) a set of parameterized policies.

2 Related work

The work presented in this paper mainly relates to two fields of research: batch mode reinforcement learning and policy gradient methods.

Genesis of batch mode RL may be found in the work of [Bradtke and Barto, 1996], where the authors use least-squares techniques in the context of temporal difference (TD) learning methods for estimating the return of control policies. This approach has been extended to the problem of optimal control by [Lagoudakis and Parr, 2003]. Algorithms similar to value iteration have also been proposed in the batch mode RL setting and the reader is referred to the works of [Ormoneit and Sen, 2002] (using kernel approximators) or [Ernst et al., 2005] (using ensembles of regression trees) and [Riedmiller, 2005] (using neural networks). More recently, new batch mode RL techniques have been proposed by [Fonteneau et al., 2013] and this does not require the use of function approximators for policy evaluation. Our policy evaluator is based on the Monte Carlo-like technique proposed by [Fonteneau et al., 2013].

Policy gradient methods [Bartlett and Baxter, 2001] can be seen as a subclass of direct policy search techniques [Schmidhuber and Zhao, 1998, Busoniu et al., 2011] that aim at finding a near-optimal policy within a set of parameterized policies. Actor-critic algorithms are relevant in this context and the reader is referred to works by [Konda and Tsitsiklis, 2003, Bhatnagar et al., 2009, Grondman et al., 2012] and the references therein. The actor-critic algorithms mentioned above work in an approximate dynamic programming setting. In other words, owing to the high-dimensional state spaces encountered often in practice, the algorithms approximate the value function with a (usually linear) function approximation architecture. Thus, the quality of the policy obtained by the algorithms are contingent upon the quality of the approximation architecture and selection of approximation architecture is in itself a hot topic of research in RL. In contrast, we employ a policy evaluation technique which does not resort to function approximation for the value function and works with a Monte Carlo like scheme instead.

3 The Setting

We consider a stochastic discrete-time system with state space $\mathcal{X} \subset \mathbb{R}^{d_{\mathcal{X}}}$, $d_{\mathcal{X}} \in \mathbb{N}$ and action space $\mathcal{U} \subset \mathbb{R}^{d_{\mathcal{U}}}$, $d_{\mathcal{U}} \in \mathbb{N}$. The dynamics of this system is governed by:

$$x_{t+1} = f(x_t, u_t, w_t), \quad \forall t \in \mathbb{N}$$

where x_t and u_t denote the state and action at time $t \in \mathbb{N}$, while $w_t \in \mathcal{W}$ denotes a random disturbance drawn according to a probability distribution $p_{\mathcal{W}}(\cdot)$. Each system transition from time t to $t + 1$ incurs an instantaneous cost $c(x_t, u_t, w_t)$. We assume that the cost function is bounded and translated into the interval $[0, 1]$.

Let $\mu : \mathcal{X} \rightarrow \mathcal{U}$ be a control policy that maps states to actions. In this paper, we consider a class of policies parameterized by $\theta \in \Theta$, i.e., $\mu^\theta : \mathcal{X} \rightarrow \mathcal{U}$. We assume that Θ is a compact and convex subset of \mathbb{R}^N , $N \in \mathbb{N}$. Since a policy μ is identifiable with its parameter θ , we shall use them interchangeably in the paper.

The classical performance criterion for evaluating a policy μ is its (expected) cost-to-go, which is the discounted sum of costs that an agent receives, while starting from a given initial state x and then following a policy μ , i.e.,

$$J^\mu(x_0) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t c(x_t, \mu(x_t), w_t) \mid x_0, \mu \right], \quad (1)$$

where $x_{t+1} = f(x_t, \mu(x_t), w_t)$ and $w_t \sim p_{\mathcal{W}}(\cdot), \forall t \in \mathbb{N}$.

In the above, $\gamma \in (0, 1)$ denotes the discount factor.

In a batch mode RL setting, the objective is to find a policy that minimizes the cost-to-go $J^\mu(x_0)$. However, the problem is challenging since the functions f , c and $p_{\mathcal{W}}(\cdot)$ are unknown (not even accessible to simulation). Instead, we are provided with a batch collection of $n \in \mathbb{N} \setminus \{0\}$ one-step system transitions \mathcal{F}_n , defined as

$$\mathcal{F}_n = \{(x^l, u^l, c^l, y^l)\}_{l=1}^n,$$

where $c^l := c(x^l, u^l, w^l)$ is the instantaneous cost and $y^l := f(x^l, u^l, w^l)$ is the next state. Here, both c^l and y^l are governed by the disturbance sequence $w^l \sim p_{\mathcal{W}}(\cdot)$, for all $l \in \{1, \dots, n\}$.

The algorithms that we present next incrementally update the policy parameter θ in the negative descent direction using either the gradient or Hessian of $J^\theta(x_0)$. The underlying policy evaluator that provides the cost-to-go inputs for any θ is based on MFMC, while the gradient/Hessian estimates are based on the principle of simultaneous perturbation (Bhatnagar et al. [2013]).

4 Algorithm Structure

In a deterministic optimization setting, an algorithm attempting to find the minima of the cost-to-go $J^\theta(x_0)$ would update the policy parameter in the descent direction as follows:

$$\theta_i(t+1) = \Gamma_i(\theta(t) - a(t)A_t^{-1}\nabla_\theta J^\theta(x_0)), \quad (2)$$

where A_t is a positive definite matrix and $a(t)$ is a step-size that satisfies standard stochastic approximation conditions: $\sum_t a(t) = \infty$ and $\sum_t a(t)^2 < \infty$. Further, $\Gamma(\theta) = (\Gamma_1(\theta_1), \dots, \Gamma_N(\theta_N))$ is a projection operator that projects the iterate θ to the nearest point in the set $\Theta \in \mathbb{R}^N$. The projection is necessary to ensure stability of the iterate θ and hence the overall convergence of the scheme (2).

For the purpose of obtaining the estimate of the cost-to-go vector $J^\theta(x_0)$ for any θ , we adapt the MFMC (for Model-Free Monte Carlo) estimator proposed by Fonteneau et al. [2010] to our (infinite-horizon discounted) setting¹. The MFMC estimator works by rebuilding (from one-step transitions taken in \mathcal{F}_n) artificial trajectories that emulate the trajectories that could be obtained if one could do Monte Carlo simulations. An estimate \hat{J}^θ of the cost-to-go J^θ is obtained by averaging the cumulative discounted cost of the rebuilt artificial trajectories.

Using the estimates of MFMC, it is necessary to build a higher-level control loop to update the parameter θ in the descent direction as given by (2). However, closed form expressions of the gradient and the Hessian of $J^\theta(x_0)$ are not available and instead, we only have (biased) estimates of $J^\theta(x_0)$ from MFMC. Thus, the requirement is for a simulation-optimization scheme that approximates the gradient/Hessian of $J^\theta(x_0)$ using estimates from MFMC.

Simultaneous perturbation methods Bhatnagar et al. [2013] are well-known simulation optimization schemes that perturb the parameter uniformly in each direction in order to find the minima of a function observable only via simulation. These methods are attractive since they require only two simulations irrespective of the parameter dimension. Our algorithms are based on two popular simultaneous perturbation methods - Simultaneous Perturbation Stochastic Approximation (SPSA) Spall [1992] and Smoothed Functional Katkovnik and Kulchitsky [1972]. The algorithms that we propose mainly differ in the choice of A_t in (2) and the specific simultaneous perturbation method used:

¹Besides being adapted to the batch mode setting, the MFMC estimator also has the advantage of having a linear computational complexity and consistency properties (see Section 5).

Algorithm 1 Structure of our algorithms.

Input: θ_0 , initial parameter vector; $\delta > 0$; Δ ;
MFMC(θ), the model free Monte Carlo like policy evaluator
for $t = 0, 1, 2, \dots$ **do**
 Call MFMC($\theta(t) + p_1(t)$)
 Call MFMC($\theta(t) + p_2(t)$)
 Compute $\theta(t+1)$ (Algorithm-specific)
end for
Return $\theta(t)$

MCPG-SPSA. Here $A_t = I$ (identity matrix). Thus, MCPG-SPSA is a first order scheme that updates the policy parameter in the descent direction. Further, the gradient $\nabla_{\theta} J^{\theta}(x_0)$ is estimated using SPSA.

MCPG-SF. This is the Smoothed functional (SF) variant of MCPG-SPSA.

MCPN-SPSA. Here $A_t = \nabla^2 J^{\theta}(x_0)$, i.e., the Hessian of the cost-to-go. Thus, MCPN is a second order scheme that update the policy parameter using a Newton step. Further, the gradient/Hessian are estimated using SPSA.

MCPN-SF. This is the SF variant of MCPN-SPSA.

As illustrated in Fig. 1, our algorithms operate on the principle of simultaneous perturbation and involve the following steps:

- (i) estimate, using MFMC, the cost-to-go for two perturbation sequences $\theta(t) + p_1(t)$ and $\theta(t) - p_2(t)$;
 - (ii) obtain the gradient/Hessian estimates (see (4)–(8)) from the cost-to-go values $J^{\theta(t)+p_1(t)}(x_0)$ and $J^{\theta(t)+p_2(t)}(x_0)$;
 - (iii) update the parameter θ in the descent direction using the gradient/Hessian estimates obtained above.
- The choice of perturbation sequences $p_1(t)$ and $p_2(t)$ is specific to the algorithm (see Sections 6.1 and 6.2).

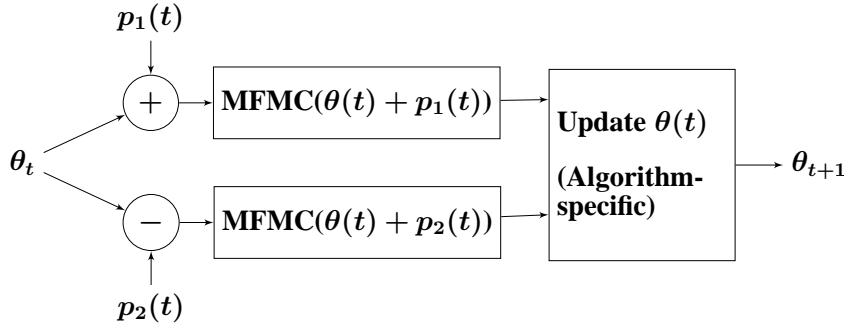


Figure 1: Overall flow of simultaneous perturbation algorithms.

Remark 1. From a theoretical standpoint, the setting considered here is of deterministic optimization and the estimates from MFMC have non-zero, albeit bounded, non-stochastic bias for a given sample of transitions. This is unlike earlier work on SPSA, which mostly feature a stochastic noise component that is zero-mean. While we establish bounds on the bias of MFMC (see Lemmas 1 and 2 in the Appendix), it is a challenge to establish asymptotic convergence and in this regard, we note the difficulties involved in Section 7.2.

5 MFMC Estimation of a Policy

For the purpose of policy evaluation given a batch of samples, we adapt the Model-free Monte Carlo estimator (MFMC) algorithm, proposed by Fonteneau et al. [2010], to an infinite horizon discounted setting.

From a sample of transitions \mathcal{F}_n , the MFMC estimator rebuilds $p \in \mathbb{N} \setminus \{0\}$ (truncated) artificial trajectories. These artificial trajectories are used as approximations of p trajectories that could be generated by simulating the policy μ^θ we want to evaluate. The final MFMC estimate $\hat{J}^\theta(x_0)$ is obtained by averaging the cumulative discounted costs over these truncated artificial trajectories.

The trajectories here are rebuilt in a manner similar to the procedure outlined by Fonteneau et al. [2010]. However, in our (infinite horizon) setting, the horizon needs to be truncated for rebuilding the trajectories. To this end, we introduce a truncation parameter T that defines the length of the rebuilt trajectories. To limit the looseness induced by such a truncation, the value of the parameter T should be chosen as a function of the discount factor γ , for instance, $T = \Omega\left(\frac{1}{1-\gamma}\right)$. The MFMC estimation can be computed using the algorithm provided in Algorithm

Algorithm 2 MFMC algorithm.

Input: $\mathcal{F}_n, \mu^\theta(\cdot, \cdot), x_0, d(\cdot, \cdot), T, p$
 \mathcal{G} : current set of not yet used one-step transitions in \mathcal{F}_n ; Initially, $\mathcal{G} \leftarrow \mathcal{F}_n$;
for $i = 1$ to p **do**
 $t \leftarrow 0$; $x_t^i \leftarrow x_0$;
 while $t < T$ **do**
 $u_t^i \leftarrow \mu^\theta(x_t^i)$;
 $\mathcal{H} \leftarrow \arg \min_{(x,u,c,y) \in \mathcal{G}} d((x,u), (x_t^i, u_t^i))$;
 $l_t^i \leftarrow$ lowest index in \mathcal{F}_n of the transitions that belong to \mathcal{H} ;
 $t \leftarrow t + 1$; $x_t^i \leftarrow y^{l_t^i}$;
 $\mathcal{G} \leftarrow \mathcal{G} \setminus \left\{ (x^{l_t^i}, u^{l_t^i}, c^{l_t^i}, y^{l_t^i}) \right\}$;
 end while
end for
Return $\hat{J}^\theta(x_0) = \frac{1}{p} \sum_{i=1}^p \sum_{t=0}^{T-1} \gamma^t c^{l_t^i}$.

2.

Definition 1 (Model-free Monte Carlo Estimator).

$$\hat{J}^\theta(x_0) = \frac{1}{p} \sum_{i=1}^p \sum_{t=0}^{T-1} \gamma^t c^{l_t^i}.$$

where $\{l_t^i\}_{i=1, t=0}^{i=p, t=T-1}$ denotes the set of indices of the transitions selected by the MFMC algorithm (see Algorithm 2).

Note that the computation of the MFMC estimator $\hat{J}^\theta(x_0)$ has a linear complexity with respect to the cardinality n of \mathcal{F}_n , the number of artificial trajectories p and the optimization horizon T .

Remark 2. Through Lemmas 1 and 2 in the Appendix, we bound the distance between the MFMC estimate $\hat{J}^\theta(x_0)$ and the true cost-to-go $J^\theta(x_0)$ in expectation and high probability, respectively.

6 Algorithms

6.1 First order algorithms

6.1.1 Gradient estimates

SPSA based estimation of the gradient of the cost-to-go is illustrated as follows: For the simple case of a scalar parameter θ ,

$$\frac{dJ^\theta}{d\theta} \approx \left(\frac{J^{\theta+\delta} - J^\theta}{\delta} \right). \quad (3)$$

The correctness of the above estimate can be seen by first $J^{\theta+\delta}$ and $J^{\theta-\delta}$ around θ using a Taylor expansion as follows:

$$J^{\theta+\delta} = J^\theta + \delta \frac{dJ^\theta}{d\theta} + O(\delta^2), J^{\theta-\delta} = J^\theta - \delta \frac{dJ^\theta}{d\theta} + O(\delta^2).$$

$$\text{Thus, } \frac{J^{\theta+\delta} - J^{\theta-\delta}}{2\delta} = \frac{dJ^\theta}{d\theta} + O(\delta).$$

From the above, it is easy to see that the estimate (3) converges to the true gradient $\frac{dJ^\theta}{d\theta}$ in the limit as $\delta \rightarrow 0$.

The above idea of simultaneous perturbation can be extended to a vector-valued parameter θ by perturbing each co-ordinate of θ uniformly using Rademacher random variables. The resulting SPSA based estimate of the gradient $\nabla_\theta J^\theta(x_0)$ is as follows:

$$\nabla_{\theta_i} J^\theta(x_0) \approx \frac{J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)}{2\delta\Delta_i}, \quad (4)$$

where $\Delta = (\Delta_1, \dots, \Delta_N)^T$ with each Δ_i being Rademacher random variables.

SF based estimation of the gradient of the cost-to-go is given by

$$\nabla_{\theta_i} J^\theta(x_0) \approx \frac{\Delta_i}{\delta} (J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)), \quad (5)$$

where Δ is a $(|N|)$ -vector of independent $\mathcal{N}(0, 1)$ random variables.

6.1.2 MCPG-SPSA and MCPG-SF algorithms

On the basis of the gradient estimate in (4)–(5), the SPSA and SF variants update the policy parameter θ as follows: For all $t \geq 1$, update

$$\text{SPSA: } \theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \frac{\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} \right), \quad (6)$$

$$\text{SF: } \theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \frac{\Delta_i(t)}{2\delta} (\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)) \right), \quad (7)$$

for all $i = 1, 2, \dots, N$. In the above,

- (i) $\delta > 0$ is a small fixed constant and $\Delta(t)$ is a N -vector of independent Rademacher random variables for SPSA and standard Gaussian random variables for SF;
- (ii) $\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0)$ and $\hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)$ are the MFMC policy evaluator's estimates of the cost-to-go corresponding to the parameters $\theta + \delta\Delta$ and $\theta - \delta\Delta$, respectively.
- (iii) $\Gamma(\theta) = (\Gamma_1(\theta_1), \dots, \Gamma_N(\theta_N))^T$ is an operator that projects the iterate θ to the closest point in a compact and convex set $\Theta \in \mathbb{R}^N$; (iv) $\{a(t), t \geq 1\}$ is a step-size sequence that satisfies the standard stochastic approximation conditions.

Remark 3. A standard approach to accelerate stochastic approximation schemes is to use Polyak-Ruppert averaging, i.e., to return the averaged iterate $\bar{\theta}_{t+1} := \sum_{s=1}^t \theta_s$ instead of θ_t .

6.2 Second order algorithms

For the second order methods, we also need an estimate of the Hessian $\nabla_\theta^2 J^\theta(x_0)$, in addition to the gradient.

6.2.1 Hessian estimates

SPSA based estimate of the Hessian $\nabla_{\theta}^2 J^{\theta}(x_0)$ is as follows:

$$\nabla_{\theta_i}^2 J^{\theta}(x_0) \approx \frac{J^{\theta+\delta\Delta+\delta\hat{\Delta}}(x_0) - J^{\theta+\delta\Delta}(x_0)}{\delta^2 \Delta_i \hat{\Delta}_i}, \quad (8)$$

where Δ and $\hat{\Delta}$ represent N -vectors of Rademacher random variables².

SF based estimate of the Hessian $\nabla_{\theta}^2 J^{\theta}(x_0)$ is as follows:

$$\nabla_{\theta_i}^2 J^{\theta}(x_0) \approx \frac{1}{\delta^2} \bar{H}(\Delta) (J^{\theta+\delta\Delta}(x_0) + J^{\theta-\delta\Delta}(x_0)), \quad (9)$$

where Δ is a N vector of independent Gaussian $\mathcal{N}(0, 1)$ random variables and $\bar{H}(\Delta)$ is a $N \times N$ matrix defined as

$$\bar{H}(\Delta) \triangleq \begin{bmatrix} (\Delta_1^2 - 1) & \Delta_1 \Delta_2 & \cdots & \Delta_1 \Delta_N \\ \Delta_2 \Delta_1 & (\Delta_2^2 - 1) & \cdots & \Delta_2 \Delta_N \\ \cdots & \cdots & \cdots & \cdots \\ \Delta_N \Delta_1 & \Delta_N \Delta_2 & \cdots & (\Delta_N^2 - 1) \end{bmatrix}. \quad (10)$$

6.2.2 MCPN-SPSA and MCPN-SF algorithms

Let $H(t) = [H_{i,j}(t)]_{i=1,j=1}^{[N],[N]}$ denote the estimate of the Hessian w.r.t. θ of the cost-to-go $J^{\theta}(x_0)$ at instant t , with $H(0) = \omega I$ for some $\omega > 0$. On the basis of (8), MCPN-SPSA would estimate the individual components $H_{i,j}(t)$ as follows: For all $t \geq 1, i, j \in \{1, \dots, N\}, i \leq j$, update

$$H_{i,j}(t+1) = H_{i,j}(t) + a(t) \left(\frac{\hat{J}^{\theta(t)+\delta\Delta(t)+\delta\hat{\Delta}(t)}(x_0) - \hat{J}^{\theta(t)+\delta\Delta(t)}(x_0)}{\delta^2 \Delta_j(t) \hat{\Delta}_i(t)} - H_{i,j}(t) \right), \quad (11)$$

and for $i > j$, set $H_{i,j}(t+1) = H_{j,i}(t+1)$. In the above, $\delta > 0$ is a small fixed constant and $\Delta(t)$ and $\hat{\Delta}(t)$ are N vectors of Rademacher random variables. Now form the Hessian inverse matrix $M(t) = \Upsilon(H(t))^{-1}$. The operator $\Upsilon(\cdot)$ ensures that the Hessian estimates stay within the set of positive definite and symmetric matrices. This is a standard requirement in second-order methods (See Gill et al. [1981] for one possible definition of $\Upsilon(\cdot)$). Using these quantities, MCPN-SPSA updates the parameter θ along a descent direction as follows: $\forall t \geq 1$,

$$\theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \sum_{j=1}^N M_{i,j}(t) \frac{\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_j(t)} \right). \quad (12)$$

Along similar lines, using (9), the SF variant of the above algorithm would update the Hessian estimate as follows: For all $t \geq 1, i, j, k \in \{1, \dots, N\}, j \leq k$, update

$$H_{i,i}(t+1) = H_{i,i}(t) + a(t) \left(\frac{(\Delta_i^2(t) - 1)}{\delta^2} (\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) + \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)) - H_{i,i}(t) \right), \quad (13)$$

$$H_{j,k}(t+1) = H_{j,k}(t) + a(t) \left(\frac{\Delta_i(t) \Delta_j(t)}{\delta^2} (\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) + \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0)) - H_{j,k}(t) \right), \quad (14)$$

and for $j > k$, we set $H_{j,k}(t+1) = H_{k,j}(t+1)$. In the above, $\Delta(t)$ is a N vector of independent Gaussian $\mathcal{N}(0, 1)$ random variables. As before, form the Hessian estimate matrix $H(t)$ and its inverse $M(t) = \Upsilon(H(t))^{-1}$. Then, the policy parameter θ is then updated as follows: $\forall t \geq 1$,

$$\theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \sum_{j=1}^N M_{i,j}(t) \Delta_j(t) \frac{(\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0))}{2\delta} \right). \quad (15)$$

²For a precise statement of the asymptotic correctness of the gradient and Hessian estimates, see Lemmas 9–10 in Appendix B.

Woodbury variant. A computationally efficient alternative to inverting the Hessian H is to use the Woodbury's identity. Woodbury's identity states that

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

where A and C are invertible square matrices and U and V are rectangular matrices of appropriate sizes. Let

$$U(t) = \frac{1}{\delta} \left[\frac{1}{\Delta_1(t)}, \frac{1}{\Delta_2(t)}, \dots, \frac{1}{\Delta_{|N|}(t)} \right]^T, V(t) = \frac{1}{\delta} \left[\frac{1}{\widehat{\Delta}_1(t)}, \frac{1}{\widehat{\Delta}_2(t)}, \dots, \frac{1}{\widehat{\Delta}_{|N|}(t)} \right] \text{ and}$$

$$C(t) = b(t) \left(\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) \right)$$

Using the Woodbury's identity, MCPN algorithm would update the estimate $M(t)$ of the Hessian inverse as follows:

$$M(t+1) = \Upsilon \left(\frac{M(t)}{1-a(t)} \left[I - \frac{C(t)U(t)V(t)M(t)}{1-b(t)+C(t)V(t)M(t)U(t)} \right] \right), \quad (16)$$

where $M(0) = kI$, with I denoting the identity matrix and k is some positive constant. The update of the policy parameter $\theta(t)$ is the same as before (see (12)).

7 Main Results

7.1 Analysis of the MFMC estimator.

(A1) We assume that the dynamics f , the cost function c and the policies $h^\theta, \forall \theta \in \Theta$ are Lipschitz continuous, i.e., we assume that there exist finite constants L_f, L_c and, $L^\theta, \forall \theta \in \Theta$ such that:

$$\forall (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W},$$

$$\|f(x, u, w) - f(x', u', w)\|_{\mathcal{X}} \leq L_f(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}),$$

$$|c(x, u, w) - c(x', u', w)| \leq L_c(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}),$$

$$\|h^\theta(x) - h^\theta(x')\|_{\mathcal{U}} \leq L^\theta\|x - x'\|_{\mathcal{X}}, \forall \theta \in \Theta,$$

where $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{U}}$ denote the chosen norms over the spaces \mathcal{X} and \mathcal{U} , respectively.

(A2) We suppose that $\mathcal{X} \times \mathcal{U}$ is bounded when measured using the distance metric d , defined as follows:

$$d((x, u), (x', u')) = \|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}},$$

$$\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2.$$

Definition 2. Given $k \in \mathbb{N} \setminus \{0\}$ with $k \leq n$, we define the k -dispersion, $\alpha_k(\mathcal{P}_n)$:

$$\alpha_k(\mathcal{P}_n) = \sup_{(x,u) \in \mathcal{X} \times \mathcal{U}} d_k^{\mathcal{P}_n}(x, u),$$

where $d_k^{\mathcal{P}_n}(x, u)$ denotes the distance of (x, u) to its k -th nearest neighbor (using the distance metric d) in the \mathcal{P}_n sample, where \mathcal{P}_n denotes the sample of state-action pairs $\mathcal{P}_n = \{(x^l, u^l)\}_{l=1}^n$.

The k -dispersion is the smallest radius such that all d -balls in $\mathcal{X} \times \mathcal{U}$ of this radius contain at least k elements from \mathcal{P}_n . We finally define the expected value of the MFMC estimator:

Definition 3 (Expected Value of $\hat{J}^\theta(x_0)$).

We denote by $E_{p, \mathcal{P}_n}^\theta(x_0)$ the expected value of the MFMC estimator that builds p trajectories:

$$E_{p, \mathcal{P}_n}^\theta(x_0) = \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\hat{J}^\theta(x_0) \right].$$

The following lemma bounds the bias of the MFMC estimator in expectation, while Lemma 2 provides a bound in high-probability.

Lemma 1. *Under (A1)-(A2), one has:*

$$\begin{aligned} |J^\theta(x_0) - E_{p, \mathcal{P}_n}^\theta(x_0)| &\leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \\ \text{with } C^\theta &= \frac{L_c}{1-\gamma L_f(1+L^\theta)} \sum_{t=0}^{T-1} \gamma^t. \end{aligned}$$

Lemma 2. *Under (A1)-(A2), one has for any $\eta > 0$:*

$$\begin{aligned} |J^\theta(x_0) - \hat{J}^\theta(x_0)| &\leq \left(C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \right) \sqrt{\frac{2 \ln(2/\eta)}{p}} \\ &\leq K_\eta \end{aligned} \tag{17}$$

with probability at least $1 - \eta$. In the above, $K_\eta > 0$ is a finite constant independent of θ .

7.2 Analysis of the MCPG algorithm

In this section, we describe the difficulty in establishing the asymptotic convergence for the MCPG-SPSA algorithm - a difficulty common to all our algorithms. An important step in the analysis is to prove that the bias in the MFMC estimator contributes a asymptotically negligible term to the θ -recursion (6). In other words, it is required to show that (6) is asymptotically equivalent to the following in the sense that the difference between the two updates is $o(1)$:

$$\theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \frac{J^{\theta(t)+\delta\Delta(t)}(x_0) - J^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} \right). \tag{18}$$

As a first step towards establishing this equivalence, we first re-write the θ -update in (6) as follows:

$$\theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \frac{J^{\theta(t)+\delta\Delta(t)}(x_0) - J^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} + a(t)\xi(t) \right),$$

where $\xi(t) = \frac{\epsilon^{\theta(t)+\delta\Delta(t)} - \epsilon^{\theta(t)-\delta\Delta(t)}}{2\delta\Delta_i(t)}$. In the above, we have used the fact MFMC returns an estimate $\hat{J}^\theta(x_0) = J^\theta(x_0) + \epsilon^\theta$, with ϵ^θ denoting the bias.

Let $\zeta(t) = \sum_{s=0}^t a(s)\xi_{s+1}$. Then, a critical requirement for establishing the equivalence of (6) with (18) is the following condition:

$$\sup_{s \geq 0} (\zeta(t+s) - \zeta(t)) \rightarrow 0 \text{ as } t \rightarrow \infty. \tag{19}$$

While the bias ϵ^θ of MFMC can be bounded (see Lemmas 1–2), it is difficult to ensure that the above condition holds.

Assuming that the bias is indeed asymptotically negligible, the asymptotic convergence of MCPG can be established in a straightforward manner. In particular, using the ordinary differential equation (ODE) approach Borkar [2008], it can be shown that (18) is a discretization (and hence converges to the equilibria) of the following ODE:

$$\dot{\theta} = \bar{\Gamma} \left(\nabla_\theta J^\theta(x_0) \right), \tag{20}$$

where $\bar{\Gamma}$ is a projection operator that ensures θ evolving according to (20) remains bounded.

Remark 4. *The detailed proof of convergence of MCPG as well as other proposed algorithms, under the assumption that (19) holds is provided in Appendix B.*

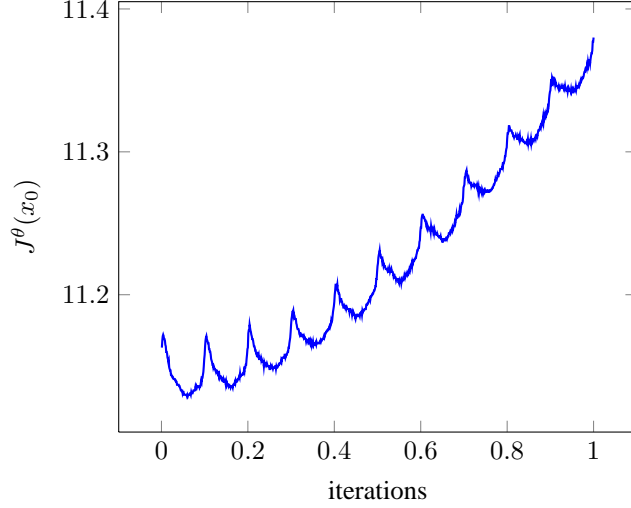


Figure 2: $J^\theta(x_0)$ vs. θ . Note that the global minimum is $\theta_{min} = 0.06$.

8 Numerical Illustration

We consider the 1-dimensional system ruled by the following dynamics:

$$f(x, u, w) = \text{sinc}(10 * (x + u + w)), \text{ where} \\ \text{sinc}(x) = \sin(\pi x) / (\pi x).$$

The cost function is defined as follows:

$$c(x, u, w) = -\frac{1}{2\pi} \exp\left(-\frac{x^2 + u^2}{2} + w\right).$$

We consider a class of linearly parameterized policies:

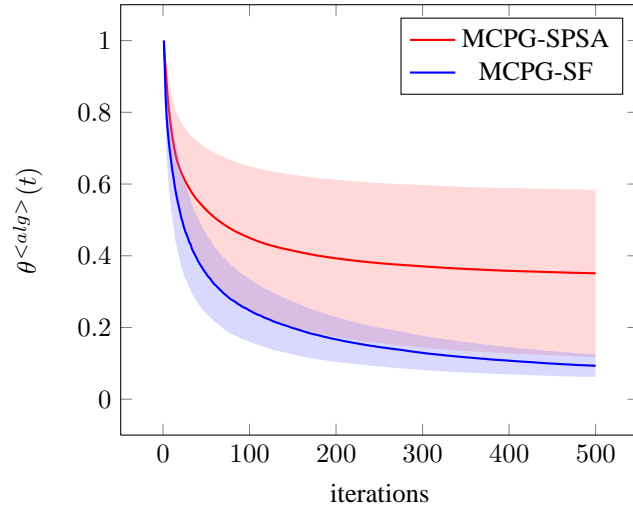
$$\mu^\theta(x) = \theta x, \quad \forall \theta \in [0, 1].$$

The disturbances are drawn according to a uniform distribution between $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ with $\epsilon = 0.01$. The initial state of the system is fixed to $x_0 = -1$ and the discount factor is set to $\gamma = 0.95$. The truncation of artificial trajectories is set to $T = \frac{1}{1-\gamma} = 20$, and the number of artificial trajectories rebuilt by the MFMC estimator is set to $p = \lceil \ln(n/T) \rceil$. We give in Figure 2 a plot of the evolution of the expected return $J^\theta(x_0)$ as a function of θ (obtained through extensive Monte Carlo simulations). We observe that the expected cost-to-go $J^\theta(x_0)$ is minimized for values of θ around 0.06.

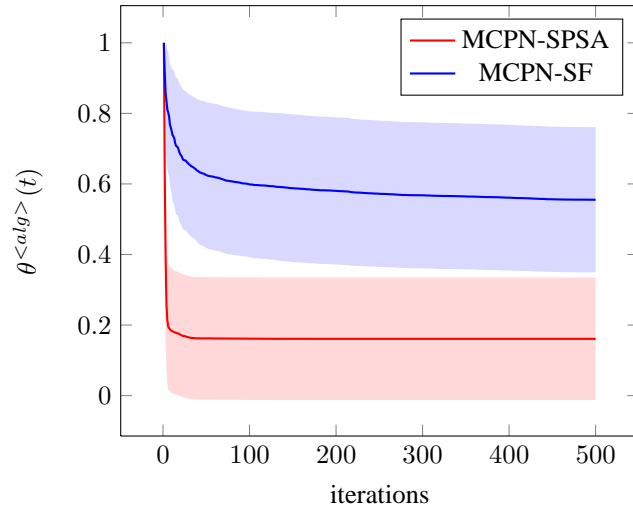
In order to observe the impact of the randomness of the set of transitions (induced by the disturbances) on the algorithms, we generate 50 samples of transitions $\mathcal{F}_n^1, \dots, \mathcal{F}_n^{50}$, each sample containing $n = 200$ transitions. For each set $\mathcal{F}_n^i, i = 1 \dots 50$, the set of state-action pairs $\mathcal{P}_n = \{(x^l, u^l)\}_{l=1}^n$ is the same and generated deterministically from a grid, i.e. $\mathcal{P}_n = \{(-1 + 2 * i / \sigma, -1 + 2 * j / \sigma)\}_{i,j=0}^{\sigma-1}$ with $\sigma = \lfloor \sqrt{n} \rfloor$. The randomness of each set \mathcal{F}_n^i comes from the disturbances w^l $l = 1 \dots n$ along which transitions are generated.

Then, for each sample \mathcal{F}_n^i , we run all the four algorithms - MCPG-SPSA, MCPG-SF, MCPN-SPSA and MCPN-SF - for 500 iterations. This generates the sequences $(\theta^{i, <alg>}(t))_t$, where $<alg>$ denotes the algorithm. For each algorithm run, we set $\delta = 0.1$ and the step-size $a(t) = \frac{1}{t}$, for all t . Further, the operator Γ projects $\theta(t)$ into the interval $[0, 1]$, while the Hessian operator Υ projects into $[0.1, \infty)$.

Figure 3 presents the average evolution of the parameter sequence in each of the 50 runs for all the algorithms (bands around the average curves represent 95% confidence intervals). From these plots, we observe that the



(a) MCPG-SPSA vs. MCPG-SF



(b) MCPN-SPSA vs. MCPN-SF

Figure 3: Empirical illustration of the MCPG and MCPN algorithms on an academic benchmark.

MCPG-SF approach outperforms the other algorithms on this academic benchmark, with a much lower variance and higher precision.

9 Extension to Risk-Sensitive Criteria

The objective here is to minimize the variance of sum of discounted costs in addition to the usual criterion of minimizing the expected cost-to-go $J^\theta(x_0)$. Recent work in this direction is by [Prashanth and Ghavamzadeh, 2013], where the authors presented actor-critic algorithms. The notable difference here is that, unlike [Prashanth and Ghavamzadeh, 2013], we use a Monte Carlo like policy evaluator and do not resort to linear function approximation for the value function. Instead, we estimate both the expected and variance of the sum of costs using a MFMC estimator and use it to solve a (constrained) risk sensitive MDP.

Let $R^\theta(x_0)$ denote the discounted sum of costs, defined as:

$$R^\theta(x_0) = \sum_{t=0}^{\infty} \gamma^t c(x_t, \mu^\theta(x_t), w_t) \quad (21)$$

with $x_{t+1} = f(x_t, \mu^\theta(x_t), w_t)$ and $w_t \sim p_W(\cdot)$. Recall that $J^\theta(x_0)$ is the expectation of this random variable. Further, let $V^\theta(x_0)$ denote the variance of $R^\theta(x_0)$. The risk-sensitive MDP, which is a constrained optimization problem, is formulated as follows:

$$\min_{\theta \in C} J^\theta(x_0) \quad \text{subject to} \quad V^\theta(x_0) \leq \alpha \quad (22)$$

In the above, $\alpha > 0$ is a constant bound on the variance that we would like to achieve. Following the technique of [Prashanth and Ghavamzadeh, 2013], we relax the above problem as $\max_\lambda \min_\theta L(\theta, \lambda) \triangleq J^\theta(x_0) + \lambda(V^\theta(x_0) - \alpha)$, where λ denotes the Lagrange multiplier.

9.1 Risk-sensitive variant of MCPG

We now describe a variant of MCPG algorithm that solves (22). The MFMC estimator is enhanced to return estimates of both the mean as well variance of the expected cost-to-go. Using these values, the algorithm would update θ and λ using a two timescale procedure as follows - (i) a faster timescale $a(t)$ for gradient descent in the primal for the θ policy parameter; (ii) a slower timescale $b(t)$ for the ascent in the dual for the Lagrange multiplier λ .

The variance $V^\theta(x_0)$ can be estimated by combining the costs given by the artificial trajectories with the classical estimator of the variance, as follows:

$$\hat{V}^\theta(x_0) = \frac{1}{p-1} \sum_{i=1}^p \left(\sum_{t=0}^{T-1} \gamma^t c^{l_i^t} - \hat{J}^\theta(x_0) \right)^2$$

We now use SPSA estimates of the gradient of the Lagrangian $L(\theta, \lambda)$ to descend in the primal and the sample of the constraint on the variance for the ascent in the Lagrange multipliers (Note: $\nabla_\lambda L(\theta, \lambda) = V^\theta(x_0) - \alpha$). This results in the following update rule for the risk-sensitive variant of MCPG algorithm:

$$\begin{aligned} \theta_i(t+1) &= \Gamma_i \left(\theta_i(t) - \frac{a(t)}{2\delta\Delta_i(t)} (\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0) + \lambda(t)(\hat{V}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{V}^{\theta(t)-\delta\Delta(t)}(x_0)) \right), \\ \lambda(t+1) &= \Gamma_\lambda \left[\lambda(t) + b(t) (\hat{V}^{\theta(t)}(x_0) - \alpha) \right]. \end{aligned} \quad (23)$$

In the above, Γ_λ is an operator that projects to $[0, \lambda_{\max}]$, where $0 < \lambda_{\max} < \infty$, while $\Gamma(\cdot)$ is the projection operator that was defined in Section 6.1.2 for the MCPG algorithm.

Remark 5. As stated by [Fonteneau et al., 2013], one can also use the MFMC estimator to output a Value-at-Risk (VaR)-like criterion as follows: Let $b \in \mathbb{R}$ and $c \in [0, 1]$.

$$\hat{J}_{RS}^{\theta, (b, c)}(x_0) = \begin{cases} +\infty & \text{if } \frac{1}{p} \sum_{i=1}^p \mathbb{I}_{\{\mathbf{c}^i > b\}} > c, \\ \hat{J}^\theta(x_0) & \text{otherwise} \end{cases}$$

where \mathbf{c}^i denotes the cost of the i -th artificial trajectory:

$$\mathbf{c}^i = \sum_{t=0}^{T-1} \gamma^t c^t.$$

This VaR-like criterion could also be optimized within the MCPG or MCPN frameworks.

9.2 Risk-sensitive variant of MCPN

We derive a variant of MCPN algorithm that incorporate the risk-related criterion of bounding the variance of the cost³. As before, we use SPSA to estimate the gradient and Hessian of the Lagrangian $L(\theta, \lambda)$. The overall update rule of this algorithm that operates on two timescales is as follows:

$$\begin{aligned} H_{i,j}(t+1) = & H_{i,j}(t) + \\ & a(t) \left(\frac{\hat{J}^{\theta(t)+\delta\Delta(t)+\delta\hat{\Delta}(t)}(x_0) - \hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) + \lambda(t) (\hat{V}^{\theta(t)+\delta\Delta(t)+\delta\hat{\Delta}(t)}(x_0) - \hat{V}^{\theta(t)+\delta\Delta(t)}(x_0))}{\delta^2 \Delta_j(t) \hat{\Delta}_i(t)} - H_{i,j}(t) \right), \end{aligned} \quad (24)$$

$$\begin{aligned} \theta_i(t+1) = & \bar{\Gamma}_i \left(\theta_i(t) - \right. \\ & a(t) \sum_{j=1}^N M_{i,j}(t) \frac{\hat{J}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{J}^{\theta(t)-\delta\Delta(t)}(x_0) + \lambda(t) (\hat{V}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{V}^{\theta(t)-\delta\Delta(t)}(x_0))}{2\delta\Delta_j(t)}, \\ & \left. - \lambda(t) \frac{\hat{V}^{\theta(t)+\delta\Delta(t)}(x_0) - \hat{V}^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} \right), \end{aligned} \quad (25)$$

$$\lambda(t+1) = \Gamma_\lambda \left[\lambda(t) + b(t) (\hat{V}^\theta(t) - \alpha) \right]. \quad (26)$$

In the above, Γ_λ is an operator that projects to $[0, \lambda_{\max}]$, where $0 < \lambda_{\max} < \infty$, while $\Gamma(\theta) = (\Gamma_1(\theta_1), \dots, \Gamma_N(\theta_N))^T$ is a projection operator that ensures θ is bounded and is the same as that used in the MCPG algorithm.

10 Conclusions

We proposed novel policy search algorithms in a batch, off-policy setting. All these algorithms incorporate simultaneous perturbation estimates for the gradient as well as the Hessian of the cost-to-go vector, since the latter is unknown and only biased estimates are available. We proposed both first order policy gradient as well as second order policy Newton algorithms, using both SPSA as well as SF simultaneous perturbation schemes. We noted certain difficulties in establishing asymptotic convergence of the proposed algorithms, owing to the non-stochastic (and non-zero) bias of the MFMC policy evaluation scheme. As a future direction, we plan to investigate conditions under which the bias of MFMC is asymptotically negligible for the policy search algorithms.

³Recall that MCPN algorithm estimated the gradient/Hessian of $J^\theta(x_0)$ alone, while not considering the variance of the return.

Appendix

A Bias and variance of the MFMC Estimator

The analysis provided in this section is an extension to the infinite horizon setting of the original analysis of the MFMC estimator [Fonteneau et al., 2010] which was done for the finite-time horizon setting. The present analysis follows the same structure.

A.1 Proof of Lemma 1

Let us first introduce the random variable $R^\theta(x_0)$ defined as follows:

$$R^\theta(x_0) = \sum_{t=0}^{\infty} \gamma^t c(x_t, \mu^\theta(x_t), w_t) \quad (27)$$

with $x_{t+1} = f(x_t, \mu^\theta(x_t), w_t)$ and $w_t \sim p_{\mathcal{W}}(\cdot)$. Before giving the proof of Lemma 1, we first give three preliminary lemmas. Given a disturbance sequence $\Omega = (\Omega(0), \Omega(1), \dots) \in \mathcal{W}^\infty$ and a policy μ^θ , we define the Ω -disturbed state-action value function $Q^{\theta, \Omega}$ as follows:

$$Q^{\theta, \Omega}(x, u) = c(x, u, \Omega(0)) + \sum_{t=1}^{\infty} \gamma^t c(x_t, \mu^\theta(x_t), \Omega(t))$$

with $x_1 = f(x, u, \Omega(0))$ and $x_{t+1} = f(x_t, \mu^\theta(x_t), \Omega(t))$, $\forall t \in \mathbb{N}$. Then, we define the expected return given Ω the quantity

$$\mathbb{E}[R^\theta(x_0)|\Omega] = \mathbb{E}[R^\theta(x_0)|w_0 = \Omega(0), w_1 = \Omega(1) \dots].$$

From there, we have the following trivial result: $\forall (\Omega, x_0) \in \mathcal{W}^\infty \times \mathcal{X}$,

$$\mathbb{E}[R^\theta(x_0)|\Omega] = Q^{\theta, \Omega}(x_0, \mu^\theta(x_0)). \quad (28)$$

Then, we have the following lemma.

Lemma 3 (Lipschitz Continuity of $Q^{\theta, \Omega}$). *Assume that $L_f(1 + L^\theta) < 1/\gamma$. Then, $\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2$,*

$$|Q^{\theta, \Omega}(x, u) - Q^{\theta, \Omega}(x', u')| \leq L_Q^\theta d((x, u), (x', u')), \text{ where } L_Q^\theta = \frac{L_c}{1 - \gamma L_f(1 + L^\theta)}.$$

Proof of Lemma 3 For the sake of conciseness, we denote $|Q^{\theta, \Omega}(x, u) - Q^{\theta, \Omega}(x', u')|$ by Δ^Q .

One has:

$$\begin{aligned} \Delta^Q &= |Q^{\theta, \Omega}(x, u) - Q^{\theta, \Omega}(x', u')| \\ &\leq |c(x, u, \Omega(0)) - c(x', u', \Omega(0))| \\ &\quad + \gamma |Q^{\theta, \Omega}(f(x, u, \Omega(0)), \mu^\theta(f(x, u, \Omega(0)))) - Q^{\mu^\theta, \Omega}(f(x', u', \Omega(0)), \mu^\theta(f(x', u', \Omega(0))))| \end{aligned}$$

and the Lipschitz continuity of c gives

$$\begin{aligned} \Delta^Q &\leq L_c d((x, u), (x', u')) + \gamma |Q^{\theta, \Omega}(f(x, u, \Omega(0)), \mu^\theta(f(x, u, \Omega(0)))) \\ &\quad - Q^{\mu^\theta, \Omega}(f(x', u', \Omega(0)), \mu^\theta(f(x', u', \Omega(0))))| \end{aligned}$$

Naming $f(x, u, \Omega(0))$ by y and $f(x', u', \Omega(0))$ by y' , we have:

$$\Delta^Q \leq L_c d((x, u), (x', u')) + \gamma |c(y, \mu^\theta(y), \Omega(1)) + \gamma Q^{\theta, \Omega}(f(y, \mu^\theta(f(y)), \Omega(1)), \mu^\theta(f(y, \mu^\theta(f(y)), \Omega(1))) - c(y', \mu^\theta(y'), \Omega(1)) - \gamma Q^{\theta, \Omega}(f(y', \mu^\theta(y'), \Omega(1)), \mu^\theta(f(y', \mu^\theta(y'), \Omega(1))))|$$

Using the Lipschitz continuity of c , we have

$$\begin{aligned} \Delta^Q &\leq L_c d((x, u), (x', u')) + \gamma L_c \Delta((y, \mu^\theta(y)), (y', \mu^\theta(y'))) \\ &\quad + \gamma^2 |Q^{\theta, \Omega} f(y, \mu^\theta(f(y)), \Omega(1)), \mu^\theta(f(y, \mu^\theta(f(y)), \Omega(1))) \\ &\quad - Q^{\theta, \Omega} f(y', \mu^\theta(f(y')), \Omega(1)), \mu^\theta(f(y', \mu^\theta(f(y')), \Omega(1)))| \end{aligned} \quad (29)$$

According to the definition of y and y' , and using the Lipschitz continuity of f and μ^θ , we have:

$$\begin{aligned} d((y, \mu^\theta(y)), (y', \mu^\theta(y'))) &= \|y - y'\|_{\mathcal{X}} + \|\mu^\theta(y) - \mu^\theta(y')\|_{\mathcal{U}} \\ &= \|f(x, u, \Omega(0)) - f(x', u', \Omega(0))\|_{\mathcal{X}} \\ &\quad + \|\mu^\theta(f(x, u, \Omega(0))) - \mu^\theta(f(x', u', \Omega(0)))\|_{\mathcal{U}} \\ &\leq L_f d((x, u), (x', u')) + L^\theta L_f d((x, u), (x', u')) \end{aligned}$$

Plugging this back in equation 29, we obtain:

$$\begin{aligned} \Delta^Q &\leq L_c d((x, u), (x', u')) + \gamma L_c (L_f d((x, u), (x', u')) + L^\theta L_f d((x, u), (x', u'))) \\ &\quad + \gamma^2 |Q^{\theta, \Omega} f(y, \mu^\theta(f(y)), \Omega(1)), \mu^\theta(f(y, \mu^\theta(f(y)), \Omega(1))) \\ &\quad - Q^{\theta, \Omega} f(y', \mu^\theta(f(y')), \Omega(1)), \mu^\theta(f(y', \mu^\theta(f(y')), \Omega(1)))| \\ &= d((x, u), (x', u')) L_c (1 + \gamma L_f (1 + L^\theta)) + \gamma^2 |Q^{\theta, \Omega} f(y, \mu^\theta(f(y)), \Omega(1)), \mu^\theta(f(y, \mu^\theta(f(y)), \Omega(1))) \\ &\quad - Q^{\theta, \Omega} f(y', \mu^\theta(f(y')), \Omega(1)), \mu^\theta(f(y', \mu^\theta(f(y')), \Omega(1)))| \end{aligned}$$

By iterating the procedure, and assuming that $L_f(1 + L^\theta) < 1/\gamma$ we obtain:

$$\begin{aligned} \Delta^Q &\leq L_c (1 + \gamma L_f (1 + L^\theta) + [\gamma L_f (1 + L^\theta)]^2 + \dots) \times d((x, u), (x', u')) \\ &= \frac{L_c}{1 - \gamma L_f (1 + L^\theta)} d((x, u), (x', u')) \end{aligned}$$

which ends the proof.

Given a truncated artificial trajectory $\tau^i = [(x^{l^i}, u^{l^i}, c^{l^i}, y^{l^i})]_{t=0}^{T-1}$ we denote by Ω^i its associated disturbance vector $\Omega^{\tau^i} = [w^{l^i_0}, \dots, w^{l^i_{T-1}}]$, i.e. the vector made of the T unknown disturbances that affected the generation of the one-step transitions $(x^{l^i_t}, u^{l^i_t}, c^{l^i_t}, y^{l^i_t})$. We give the following lemma.

Lemma 4 (Bounds on the expected return given Ω). $\forall i \in \{1, \dots, p\}$,

$$b^\theta(\tau^i, x_0) \leq \mathbb{E}[R^\theta(x_0) | \Omega^i] \leq a^\theta(\tau^i, x_0),$$

with

$$\begin{aligned} b^\theta(\tau^i, x_0) &= \sum_{t=0}^{T-1} \gamma^t [c^{l^i_t} - L_Q^\theta \psi_t^i] - \frac{\gamma^T}{1 - \gamma}, \\ a^\theta(\tau^i, x_0) &= \sum_{t=0}^{T-1} \gamma^t [c^{l^i_t} + L_Q^\theta \psi_t^i] + \frac{\gamma^T}{1 - \gamma}, \\ \psi_t^i &= d((x^{l^i_t}, u^{l^i_t}), (y^{l^i_{t-1}}, \mu^\theta(y^{l^i_{t-1}}))), \forall t \in \{0, \dots, T-1\}, \\ y^{l^i_{-1}} &= x_0, \forall i \in \{1, \dots, p\}. \end{aligned}$$

Proof of Lemma 4 Let us first prove the lower bound. With $u_0 = \mu^\theta(x_0)$, the Lipschitz continuity of $Q^{\theta, \Omega^{\tau^i}}$ gives

$$|Q^{\theta, \Omega^i}(x_0, u_0) - Q^{\theta, \Omega^i}(x^{l_0^i}, u^{l_0^i})| \leq L_Q^\theta d((x_0, u_0), (x^{l_0^i}, u^{l_0^i})) .$$

Equation (28) gives

$$Q^{\theta, \Omega^i}(x_0, u_0) = \mathbb{E}[R^\theta(x_0)|\Omega^i].$$

Thus,

$$\begin{aligned} |\mathbb{E}[R^\theta(x_0)|\Omega^i] - Q^{\theta, \Omega^i}(x^{l_0^i}, u^{l_0^i})| &= |Q^{\theta, \Omega^i}(x_0, \mu^\theta(x_0)) - Q^{\theta, \Omega^{\tau^i}}(x^{l_0^i}, u^{l_0^i})| \\ &\leq L_Q^\theta d((x_0, \mu^\theta(x_0)), (x^{l_0^i}, u^{l_0^i})) . \end{aligned} \quad (30)$$

It follows that

$$Q^{\theta, \Omega^i}(x^{l_0^i}, u^{l_0^i}) - L_Q^\theta \psi_0^i \leq \mathbb{E}[R^\theta(x_0)|\Omega^i] .$$

Then, we know that

$$Q^{\theta, \Omega^i}(x^{l_0^i}, u^{l_0^i}) = c(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}) + \gamma Q^{\theta, \Omega^i}(f(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}), \mu^\theta(f(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}))) .$$

By definition of Ω^i , we have: $c(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}) = c^{l_0^i}$ and $f(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}) = y^{l_0^i}$. From there

$$Q^{\theta, \Omega^i}(x^{l_0^i}, u^{l_0^i}) = c^{l_0^i} + \gamma Q^{\theta, \Omega^i}(y^{l_0^i}, \mu^\theta(y^{l_0^i})) ,$$

and

$$\gamma Q^{\theta, \Omega^i}(y^{l_0^i}, \mu^\theta(y^{l_0^i})) + c^{l_0^i} - L_Q^\theta \psi_0^i \leq \mathbb{E}[R^\theta(x_0)|\Omega^i] .$$

The Lipschitz continuity of Q^{θ, Ω^i} gives

$$|Q^{\theta, \Omega^i}(y^{l_0^i}, \mu^\theta(y^{l_0^i})) - Q^{\theta, \Omega^i}(x^{l_1^i}, u^{l_1^i})| \leq L_Q^\theta d((y^{l_0^i}, \mu^\theta(y^{l_0^i})), (x^{l_1^i}, u^{l_1^i})) = L_Q^\theta \psi_1^i ,$$

which implies that

$$L_Q^\theta \psi_1^i \leq Q^{\theta, \Omega^i}(y^{l_0^i}, \mu^\theta(y^{l_0^i})) .$$

We therefore have

$$\gamma Q^{\theta, \Omega^i}(x^{l_1^i}, u^{l_1^i}) + c^{l_0^i} - L_Q^\theta \psi_0^i - \gamma L_Q^\theta \psi_1^i \leq \mathbb{E}[R^\theta(x_0)|\Omega^i] .$$

The proof is completed by iterating this derivation, and by bounding the uncertainty induced by the truncation, which adds a term $\frac{\gamma^T}{1-\gamma}$ to the bound since the reward function c takes value in $[0, 1]$. The upper bound is proved similarly. We give a third lemma.

Lemma 5. $\forall i \in \{1, \dots, p\}$,

$$a^\theta(\tau^i, x_0) - b^\theta(\tau^i, x_0) \leq 2 \left(C \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \right)$$

with $C^\theta = L_Q^\theta \sum_{t=0}^{T-1} \gamma^t$.

Proof of Lemma 5 By construction of the bounds, one has $a^\theta(\tau^i, x_0) - b^\theta(\tau^i, x_0) = \sum_{t=0}^{T-1} 2\gamma^t L_Q^\theta \psi_t^i + \frac{2\gamma^T}{1-\gamma}$. The MFMC algorithm chooses $p \times T$ different one-step transitions to build the MFMC estimator by minimizing the distance $d((y^{l_{t-1}^i}, \mu^\theta(y^{l_{t-1}^i})), (x^{l_t^i}, u^{l_t^i}))$, so by definition of the k -sparsity of the sample \mathcal{P}_n with $k = pT$, one has

$$\psi_t^i = d((y^{l_{t-1}^i}, \mu^\theta(y^{l_{t-1}^i})), (x^{l_t^i}, u^{l_t^i})) \leq d_{pT}^{\mathcal{P}_n}(y^{l_{t-1}^i}, \mu^\theta(y^{l_{t-1}^i})) \leq \alpha_{pT}(\mathcal{P}_n) ,$$

which ends the proof.

Using those three lemmas, one can now compute an upper bound on the bias of the MFMC estimator.

Proof of Lemma 1 By definition of $a^\theta(\tau^i, x_0)$ and $b^\theta(\tau^i, x_0)$, we have

$$\forall i \in \{1, \dots, p\}, \frac{b^\theta(\tau^i, x_0) + a^\theta(\tau^i, x_0)}{2} = \sum_{t=0}^{T-1} \gamma^t c^{l_i^i}.$$

Then, according to Lemmas 4 and 5, we have $\forall i \in \{1, \dots, p\}$,

$$\begin{aligned} \left| \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\mathbb{E}[R^\theta(x_0) | \Omega^i] - \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} \right] \right| &\leq \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\left| \mathbb{E}[R^\theta(x_0) | \Omega^i] - \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} \right| \right] \\ &\leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma}. \end{aligned}$$

Thus,

$$\begin{aligned} \left| \frac{1}{p} \sum_{i=1}^p \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\mathbb{E}[R^\theta(x_0) | \Omega^i] - \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} \right] \right| &\leq \frac{1}{p} \sum_{i=1}^p \left| \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\mathbb{E}[R^\theta(x_0) | \Omega^i] - \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} \right] \right| \\ &\leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma}, \end{aligned}$$

which can be reformulated

$$\left| \mathbb{E}_{w^1, \dots, w^n \sim p_{\mathcal{W}}(\cdot)} \left[\frac{1}{p} \sum_{i=1}^p \mathbb{E}[R^\theta(x_0) | \Omega^i] \right] - E_{p, \mathcal{P}_n}^\theta(x_0) \right| \leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma},$$

since $\frac{1}{p} \sum_{i=1}^p \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} = \hat{J}^\theta(x_0)$. Since the MFMC algorithm chooses $p \times T$ different one-step transitions, all the $\{w^{l_i^i}\}_{i=1, t=0}^{p, T-1}$ are i.i.d. according to $p_{\mathcal{W}}(\cdot)$. For all $i \in \{1, \dots, p\}$, The law of total expectation gives

$$\mathbb{E}_{w^{l_0^i}, \dots, w^{l_{T-1}^i} \sim p_{\mathcal{W}}(\cdot)} \left[\mathbb{E}_{w^{l_0^i}, \dots, w^{l_{T-1}^i} \sim p_{\mathcal{W}}(\cdot)} [R^\theta(x_0) | \Omega^i] \right] = \mathbb{E}_{w_0, \dots, w_{T-1} \sim p_{\mathcal{W}}(\cdot)} [R^\theta(x_0)] = J^\theta(x_0).$$

This ends the proof.

A.2 Proof of Lemma 2

One first have the triangle inequality.

$$\left| \hat{J}^\theta(x_0) - J^\theta(x_0) \right| \leq \left| \hat{J}^\theta(x_0) - \frac{1}{p} \sum_{i=1}^p \mathbb{E}[R^\theta(x_0) | \Omega^i] \right| + \left| \frac{1}{p} \sum_{i=1}^p \mathbb{E}[R^\theta(x_0) | \Omega^i] - J^\theta(x_0) \right|.$$

From the proof given above, one has the following property: $\forall i \in \{1, \dots, p\}$,

$$\left| \sum_{t=0}^{T-1} \gamma^t c^{l_i^i} - \mathbb{E}[R^\theta(x_0) | \Omega^i] \right| \leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma}. \quad (31)$$

This immediatly leads to:

$$\left| \hat{J}^\theta(x_0) - \frac{1}{p} \sum_{i=1}^p \mathbb{E}[R^\theta(x_0) | \Omega^i] \right| \leq C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma}.$$

From Equation (31), we have that each variable $\mathbb{E} [R^\theta(x_0) | \Omega^i]$ is contained in the interval

$$\left[\sum_{t=0}^{T-1} \gamma^t c^{l_t^i} - C^\theta \alpha_{pT}(\mathcal{P}_n) - \frac{\gamma^T}{1-\gamma}, \sum_{t=0}^{T-1} \gamma^t c^{l_t^i} + C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \right]$$

of width $2 \left(C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \right)$ with probability one. Since all $\{w^{l_t^i}\}, i = 1 \dots p, t = 0 \dots T-1$ are i.i.d. from $p_{\mathcal{W}}(\cdot)$, we can apply the Chernoff-Hoeffding inequality:

$$\left| \frac{1}{p} \sum_{i=1}^p \mathbb{E} [R^\theta(x_0) | \Omega^i] - J^\theta(x_0) \right| = |\hat{J}^\theta(x_0) - J^\theta(x_0)| \leq \left(C^\theta \alpha_{pT}(\mathcal{P}_n) + \frac{\gamma^T}{1-\gamma} \right) \sqrt{\frac{2 \ln(2/\eta)}{p}}$$

with probability at least $1 - \eta$. The proof of Equation 17 is obtained by observing that there exists a constant $C := \sup_\theta C^\theta < \infty$. The existence of $C < \infty$ is ensured by the fact that (i) μ^θ is continuously differentiable function of θ and (ii) θ evolves within a compact set, so the Lipschitz constant of any policy θ is finite.

B Asymptotic convergence of the policy gradient methods

We make the following assumptions for the analysis:

(A3) The policy μ^θ is continuously differentiable for any policy parameter $\theta \in \Theta$.

(A4) The underlying Markov chain corresponding to any policy θ is irreducible and positive recurrent.

(A5) The step-size sequence $a(n)$ satisfies

$$\sum_{n=1}^{\infty} a(n) = \infty \text{ and } \sum_{n=1}^{\infty} a(n)^2 < \infty.$$

(A6) The bias of MFMC satisfies the following condition:

$$\text{Let } \zeta(t) = \sum_{s=0}^t a(s) \xi_{s+1}, \text{ then } \sup_{s \geq 0} (\zeta(t+s) - \zeta(t)) \rightarrow 0 \text{ as } t \rightarrow \infty.$$

The first assumption is standard in policy gradient RL algorithms, while the second assumption ensures that each state gets visited an infinite number of times over an infinite time horizon. The third assumption above imposes standard stochastic approximation conditions on the step-sizes, while the final assumption ensures that the bias of MFMC is asymptotically negligible.

B.1 Analysis of MCPG-SPSA

Before we proceed with the analysis of MCPG, we re-state the following fact regarding the bias of the estimate returned by MFMC: Let ϵ^θ denote the bias of the MFMC estimate $\hat{J}^\theta(x_0)$, i.e., $\hat{J}^\theta(x_0) = J^\theta(x_0) + \epsilon^\theta$. Then, the bias ϵ^θ satisfies the following bound:

$$\forall \theta \in \Theta, \|\epsilon^\theta\|_2 \leq K_\eta \text{ with probability at least } 1 - \eta. \quad (32)$$

for some positive, finite constant K_η independent from θ . Fix $\eta > 0$ and let E^η denote the set of all θ on which (32) holds, i.e., $E^\eta = \{\theta \in \Theta \mid \|\epsilon^\theta\|_2 \leq K_\eta\}$.

We use the ordinary differential equation (ODE) approach (Borkar [2008]) to analyze our algorithms. Under (A6), the update rule (6) of MCPG can be seen to be asymptotically equivalent to⁴:

$$\theta_i(t+1) = \Gamma_i \left(\theta_i(t) - a(t) \frac{J^{\theta(t)+\delta\Delta(t)}(x_0) - J^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} \right). \quad (33)$$

⁴the equivalence is in the sense that the difference between the (6) and (33) is $o(1)$.

The proof of convergence of the first order method MCPG is to a set of asymptotically stable equilibrium points of the following ODE:

$$\dot{\theta} = \bar{\Gamma}(\nabla_{\theta} J^{\theta}(x_0)). \quad (34)$$

In the above, $\bar{\Gamma}$ is a projection operator that is defined as follows: For any bounded continuous function $g(\cdot)$,

$$\bar{\Gamma}(g(\theta)) = \lim_{\tau \rightarrow 0} \frac{\Gamma(\theta + \tau g(\theta)) - \theta}{\tau}. \quad (35)$$

The projection operator $\bar{\Gamma}(\cdot)$ is necessary to ensure that θ , while evolving through the ODE (34), stays within the bounded set $\Theta \in \mathbb{R}^N$. Let $\mathcal{Z} = \{\theta \in C : \bar{\Gamma}(\nabla J^{\theta}(x_0)) = 0\}$ denote the set of asymptotically stable equilibria of the ODE (34). The main result regarding the convergence of MCPG is as follows:

Theorem 6. *Under (A1)-(A6), for any $\eta > 0$, $\theta(t)$ governed by (6) converges to \mathcal{Z} in the limit as $\delta \rightarrow 0$, with probability $1 - \eta$.*

Before proving Theorem 6, we prove that the correctness of the SPSA-based gradient estimate (4) in the following lemma⁵:

Lemma 7. *Recall that $\Delta = (\Delta_1, \dots, \Delta_N)^T$ is vector of independent Rademacher random variables. We have*

$$\lim_{\delta \rightarrow 0} \frac{J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)}{2\delta\Delta_i(t)} = \nabla_i J^{\theta}(x_0). \quad (36)$$

Proof. Using a Taylor expansion of $J^{\theta+\delta\Delta}(x_0)$ and $J^{\theta-\delta\Delta}(x_0)$ around θ , we obtain:

$$J^{\theta(t)+\delta\Delta(t)}(x_0) = J^{\theta(t)}(x_0) + \delta\Delta(t)^T \nabla J^{\theta(t)}(x_0) + O(\delta^2), \quad (37)$$

$$J^{\theta(t)-\delta\Delta(t)}(x_0) = J^{\theta(t)}(x_0) - \delta\Delta(t)^T \nabla J^{\theta(t)}(x_0) + O(\delta^2). \quad (38)$$

From the above, it is easy to see that

$$\frac{J^{\theta(t)+\delta\Delta(t)}(x_0) - J^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)} - \nabla_i J^{\theta(t)}(x_0) \quad (39)$$

$$= \underbrace{\sum_{j=1, j \neq i}^N \frac{\Delta_j(t)}{\Delta_i(t)} \nabla_j J^{\theta(t)}(x_0)}_{(I)} + O(\delta) \quad (40)$$

Term (I) above is zero since Δ are Rademacher. So, it is easy to see that the estimate (36) converges to the true gradient $\nabla J^{\theta(t)}(x_0)$ in the limit as $\delta \rightarrow 0$. \square

Proof. (Theorem 6) In lieu of (A6), it is sufficient to analyse the following equivalent update rule for MCPG on the high-probability set E^{η} :

$$\theta_i(t+1) = \Gamma_i\left(\theta_i(t) - a(t) \frac{J^{\theta(t)+\delta\Delta(t)}(x_0) - J^{\theta(t)-\delta\Delta(t)}(x_0)}{2\delta\Delta_i(t)}\right).$$

Now, using a standard Taylor series expansion (see Chapter 5 of [Bhatnagar et al., 2013]) it is easy to show that $\frac{J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)}{2\delta\Delta_i(t)}$ is a biased estimator of $\nabla_{\theta} J^{\theta}(x_0)$, where the bias vanishes asymptotically. In more rigorous terms, we have

$$\frac{J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)}{2\delta\Delta_i(t)} \xrightarrow{\delta \rightarrow 0} \nabla_{\theta_i} J^{\theta}(x_0).$$

⁵The proof is given here for the sake of completeness and the reader is referred to Chapter 5 of Bhatnagar et al. [2013] for an extensive treatment on SPSA based gradient estimation.

Thus, Eq. 6 can be seen to be a discretization of the ODE (34). Further, \mathcal{Z}_λ is an asymptotically stable attractor for the ODE (34), with $J^\theta(x_0)$ itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{dJ^\theta(x_0)}{dt} = \nabla_\theta J^\theta(x_0) \dot{\theta} = \nabla_\theta J^\theta(x_0) \bar{\Gamma} (-\nabla_\theta J^\theta(x_0)) < 0.$$

The claim now follows from Theorem 5.3.3, pp. 191-196 of [Kushner and Clark, 1978]. Note that the final claim holds on E^η , the set with high probability on which the bias of the MFMC estimator is bounded. \square

B.2 Convergence analysis of MCPN-SPSA

We establish that policy parameter θ governed by MCPN algorithm (12) converges to the set of asymptotically stable equilibria of the following ODE:

$$\dot{\theta} = \bar{\Gamma} ((\nabla_\theta^2 J^\theta(x_0))^{-1} \nabla_\theta J^\theta(x_0)). \quad (41)$$

In the above, $\bar{\Gamma}$ is as defined in (35). Let $\mathcal{Z} = \{\theta \in C : \bar{\Gamma}((\nabla_\theta^2 J^\theta(x_0))^{-1}) = 0\}$ denote the set of asymptotically stable equilibria of the ODE (41).

The main result regarding the convergence of $\theta(t)$ governed by (12) is given as follows:

Theorem 8. *Under (A1)-(A6), for any $\eta > 0$, $\theta(t)$ governed by (12) converges to \mathcal{Z} in the limit as $\delta \rightarrow 0$, with probability $1 - \eta$.*

Before we prove Theorem 8, we establish that the Hessian estimate $H(t)$ in (12) converges almost surely to the true Hessian $\nabla_\theta^2 J^\theta(x_0)$ in the following lemma.

Lemma 9. *With $\delta \rightarrow 0$ as $t \rightarrow \infty$, for all $i, j \in \{1, \dots, N\}$, we have the following claims with probability one:*

$$(i) \left\| \frac{J^{\theta(t)+\delta\Delta(t)+\delta\hat{\Delta}(t)}(x_0) - J^{\theta(t)+\delta\Delta(t)}(x_0)}{\delta^2 \Delta_i(t) \hat{\Delta}_j(t)} - \nabla_{i,j}^2 J^{\theta(t)}(x_0) \right\| \rightarrow 0,$$

$$(ii) \|H_{i,j}(t) - \nabla_{i,j}^2 J^{\theta(t)}(x_0)\| \rightarrow 0,$$

$$(iii) \|M(t) - \Upsilon(\nabla^2 J^{\theta(t)}(x_0))^{-1}\| \rightarrow 0.$$

Proof. The above claims can be established by employing standard Taylor series expansions. For a detailed derivation, the reader is referred to Propositions 7.12 and Lemmas 7.10 and 7.11 of [Bhatnagar et al., 2013], respectively. \square

Proof. (Theorem 8) As in the case of the first order method, we can use (A6) to arrive at the following update rule equivalent of the policy parameter θ on the high-probability set E^η :

$$H_{i,j}(t+1) = H_{i,j}(t) + a(t) \left(\frac{J^{\theta+\delta\Delta+\delta\hat{\Delta}}(x_0) - J^{\theta+\delta\Delta}(x_0)}{\delta^2 \Delta_j(t) \hat{\Delta}_i(t)} - H_{i,j}(t) \right), \quad (42)$$

$$\theta_i(t+1) = \bar{\Gamma}_i \left(\theta_i(t) + a(t) \sum_{j=1}^N M_{i,j}(t) \frac{J^{\theta-\delta\Delta}(x_0) - J^{\theta+\delta\Delta}(x_0)}{\delta \hat{\Delta}_j(t)} \right), \quad (43)$$

In lieu of Lemma 9, it can be seen that $H_{i,j}(t)$ converges to the true Hessian $\nabla_{\theta_i}^2 J^\theta(x_0)$ as $\delta \rightarrow 0$. Thus, the θ -recursion above is equivalent to the following on E^η :

$$\theta_i(t+1) = \bar{\Gamma}_i \left(\theta_i(t) + a(t) (\nabla_{\theta_i}^2 J^\theta(x_0))^{-1} \nabla_{\theta_i} J^\theta(x_0) \right). \quad (44)$$

The above can be seen as a discretization of the ODE (41). Thus, the $\theta(t)$ governed by (12) can be seen to converge to a set containing the asymptotically stable equilibria of the above ODE, albeit with probability $1 - \eta$ for any $\eta > 0$. \square

B.3 Analysis of SF-based algorithms - MCPG-SF and MCPN-SF

One can prove SF variants of Theorems 6 and 8 along similar lines, using the following lemma: Recall that Δ is a N -vector of independent Gaussian $\mathcal{N}(0, 1)$ random variables for SF-based algorithms.

Lemma 10. *With $\delta \rightarrow 0$ as $t \rightarrow \infty$, for all $i, j \in \{1, \dots, N\}$, we have the following claims with probability one: (The expectations in the following are w.r.t. the distribution of perturbation random variables Δ)*

$$(i) \left\| \mathbb{E} \left[\frac{\Delta_i}{\delta} (J^{\theta+\delta\Delta}(x_0) - J^{\theta-\delta\Delta}(x_0)) \right] - \nabla_i J^\theta(x_0) \right\| \rightarrow 0,$$

$$(ii) \left\| E \left[\frac{1}{\delta^2} \bar{H}(\Delta)(J^{\theta+\delta\Delta}(x_0) + J^{\theta-\delta\Delta}(x_0)) \right] - \nabla_{i,j}^2 J^\theta(x_0) \right\| \rightarrow 0,$$

Proof. The proofs of the above claims follow from Propositions 6.5 and 8.10 of Bhatnagar et al. [2013], respectively. \square

References

- P. L. Bartlett and J. Baxter. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.
- D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3)*. Athena Scientific, May 1996.
- S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11): 2471–2482, 2009.
- S. Bhatnagar, Prasad H.L., and Prashanth L.A. *Stochastic Recursive Algorithms for Optimization*, volume 434. Springer, 2013.
- V.S. Borkar. *Stochastic Approximation: a Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- S.J. Bradtke and A.G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.
- L. Busoniu, D. Ernst, B. De Schutter, and R. Babuska. Cross-entropy optimization of control policies with adaptive basis functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(1):196–209, 2011.
- D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.
- R. Fonteneau. Contributions to Batch Mode Reinforcement Learning. *PhD Thesis, University of Liège*, 2011.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo-like policy evaluation. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 217–224, 2010.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Batch mode reinforcement learning based on the synthesis of artificial trajectories. *Annals of Operations Research*, 208:383–416, 2013.
- P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic press, 1981.
- I. Grondman, L. Busoniu, G. AD Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1291–1307, 2012.

- V. Katkovnik and Y. Kulchitsky. Convergence of a class of random search algorithms. *Automatic Remote Control*, 8:81–87, 1972.
- Vijay R Konda and John N Tsitsiklis. On actor-critic algorithms. *SIAM journal on Control and Optimization*, 42(4):1143–1166, 2003.
- H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978. ISBN 0-387-90341-0.
- M.G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.
- D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.
- L.A. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.
- M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328, 2005.
- J. Schmidhuber and J. Zhao. Direct policy search and uncertain policy evaluation. Technical report, In AAAI Spring Symposium on Search under Uncertain and Incomplete Information, Stanford Univ, 1998.
- J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. ISSN 0018-9286.